Audio Engineering Society

Conference Paper 7

Presented at the AES International Conference on Machine Learning and Artificial Intelligence for Audio
2025 September 8–10, London, UK

This paper was peer-reviewed as a complete manuscript for presentation at this conference. This paper is available in the AES E-Library (http://www.aes.org/e-lib), all rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

Automatic Audio Equalization with Semantic Embeddings

Eloi Moliner[1], Vesa Välimäki[1], Konstantinos Drossos[2], and Matti S. Hämäläinen[2]

[1]*Acoustics Lab, Department of Information and Communications Engineering, Aalto University, Espoo, Finland*
[2]*Nokia Technologies, Tampere/Espoo, Finland*

Correspondence should be addressed to Eloi Moliner (eloi.moliner@aalto.fi)

ABSTRACT

This paper presents a data-driven approach to automatic blind equalization of audio by predicting log-mel spectral features and deriving an inverse filter. The method uses a deep neural network, where a pre-trained model provides semantic embeddings as a backbone, and only a lightweight head is trained. This design is intended to enhance training efficiency and generalization. Trained on both music and speech, the model is robust to noise and reverberation. Objective evaluations confirm its effectiveness, and subjective tests show performance comparable to that of an oracle that uses true log-mel spectral features, indicating that the model accurately estimates the desired characteristics, with remaining limitations attributed to the filtering stage. Overall, the results highlight the potential of the method for real-world audio enhancement applications.

1 Introduction

Automatic blind equalization (EQ) of speech and music is a key aspect in audio processing. The goal is to adjust the spectrum of an audio signal to achieve a desired tonal balance without knowledge of the original recording conditions or the target equalization curve [1]. Such processing enhances the audio listening experience by improving clarity and consistency. By enabling data-driven adjustments, EQ can significantly improve sound quality across a wide range of contexts. Automatic equalization is used in music production [2, 3], hearing aids [4], and teleconferencing [5].

This problem has been studied extensively. A well-known approach involves estimating an equalization curve by averaging the spectral characteristics of a reference recording or collection of recordings [6, 7]. More recently, deep learning techniques have been explored, employing neural networks to model complex example-dependent spectral transformations. End-to-end convolutional networks have been applied to automatic EQ [8], while methods based on self-supervised frameworks have been proposed for blind parameter estimation [9, 10, 11]. Closer to this work, Mockenhaupt et al. [12] estimated EQ targets using a classification-based approach, though the reliance on manually defined targets limits flexibility. Generative models have also been explored, with Moliner et al. applying a diffusion-based model for equalization and audio restoration [13].

While deep learning has shown promise for EQ, practical deployment requires models that generalize across diverse signals and remain robust to noise and reverberation. This paper presents a data-driven approach to blind equalization by predicting log-mel spectral features and deriving an inverse filter, explicitly optimized for robustness and generalization.

This paper is organized as follows. Sec. 2 defines the blind EQ problem as an inverse filtering task. Sec. 3 details the proposed method, which combines a pre-trained feature extractor with a lightweight trainable head for parameter estimation. Sec. 4 describes the experimentals, including training on speech and music. Secs. 5 and 6 present objective and subjective evaluations. Finally, Sec. 7 summarizes key findings.

2 Problem Definition

The problem is how to automatically equalize the tonal balance of a degraded audio signal towards a reference. Let $\mathbf{x} \in \mathbb{R}^L$ represent a high-quality reference audio signal. We observe a degraded version of this signal, $\mathbf{y} \in \mathbb{R}^L$, generated through the forward model:

$$\mathbf{y} = \mathcal{A}(\mathbf{x}) + \mathbf{n}, \tag{1}$$

where $\mathcal{A}(\cdot) : \mathbb{R}^L \to \mathbb{R}^L$ is a degradation operator (e.g., filtering or nonlinear distortion), and $\mathbf{n} \in \mathbb{R}^L$ captures additive noise from environmental interference, equipment limitations, or other sources. Our objective is to estimate a reconstruction $\hat{\mathbf{x}} \approx \mathbf{x}$. Assuming that $\mathcal{A}(\cdot)$ is a linear time-invariant system, the degradation can be modeled as a convolution with a finite impulse response filter $\mathbf{h} \in \mathbb{R}^K$ of length K:

$$\mathcal{A}(\mathbf{x}) \approx \mathbf{h} * \mathbf{x}, \tag{2}$$

where $*$ denotes discrete convolution. We further assume that the problem is well-conditioned, implying that the filter is invertible, with inverse $\mathbf{h}^{-1}$. In the noiseless case, the reference can be approximated as

$$\mathbf{x} \approx \hat{\mathbf{x}} = \hat{\mathbf{h}}^{-1} * \mathbf{y}. \tag{3}$$

Thus, the task of estimating the reference is reduced to estimating the inverse filter $\hat{\mathbf{h}}^{-1}$. However, this approximation does not hold in the presence of noise $\mathbf{n}$. As we will explain later, this issue is addressed by pre-processing $\mathbf{y}$ with a denoiser. In cases where the problem is ill-conditioned, such as when portions of the signal (e.g., certain frequency bands or segments) are lost or severely degraded, our method is limited to restoring only the nondegraded parts of the signal.

3 Methods

3.1 Approximating the Inverse Filter

This work aims to estimate the inverse filter $\mathbf{h}_i^{-1}$ without any prior knowledge of the forward filter $\mathbf{h}_i$, making it a blind estimation task. The approach is performed in the frequency domain, where we estimate the inverse filter $\hat{\mathbf{H}}^{-1} \in \mathbb{C}^{N_{\text{FFT}}}$, based on a chosen FFT size N_{FFT}.

Equation 3 can be written in the frequency domain as

$$\mathbf{x} \approx \mathcal{F}^{-1}\left(\hat{\mathbf{H}}^{-1} \odot \mathcal{F}(\mathbf{y})\right), \tag{4}$$

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote the short-time Fourier transform (STFT) and its inverse (iSTFT), respectively. Both transformations are computed using N_{FFT} frequency bins, with a window size and hop size defined in Sec. 4.3. Owing to the Hermitian symmetry of the Fourier transform for real-valued signals, only $N_{\text{FFT}}/2+1$ bins contain non-redundant information, and thus only these need to be estimated in practice.

We constrain the inverse filter $\hat{\mathbf{H}}^{-1}$ to be a zero-phase filter, and approximate its magnitude as

$$|\hat{\mathbf{H}}^{-1}| = \sqrt{\frac{\mathbf{X}^2_{\text{ref}}}{\mathbf{Y}^2_{\text{avg}}}}, \tag{5}$$

where $\mathbf{Y}^2_{\text{avg}}$ is the time-averaged Power Spectral Density (PSD) of the observed signal $\mathbf{y}$, computed as

$$\mathbf{Y}^2_{\text{avg}} = \frac{1}{M}\sum_{m=0}^{M-1} |\text{STFT}(\mathbf{y})_m|^2. \tag{6}$$

Here, M is the number of STFT frames, and $\mathbf{X}^2_{\text{ref}}$ denotes the target or reference spectral shape, which is typically content-dependent. The reference $\mathbf{X}^2_{\text{ref}}$ could be a user-defined spectral curve, or, more generally, the average power spectrum of the unknown clean source $\mathbf{x}$, given in an analogous way as in Eq. (6).

Since the clean source $\mathbf{x}$ is unknown, we estimate the target spectrum $\hat{\mathbf{X}}^2_{\text{ref}} \approx \mathbf{X}^2_{\text{ref}}$ using a deep neural network (DNN). The DNN denoted as $E_\theta(\mathbf{y})$ is trained to predict the target spectrum given the observed signal $\mathbf{y}$, where θ represents the trainable parameters of the network.

Instead of predicting the target spectral shape directly $\hat{\mathbf{X}}^2_{\text{ref}}$, we reduce its dimensionality by using a mel-scale triangular filterbank. The mel-transformed reference spectrum $\mathbf{z}_{\text{ref}} \in \mathbb{R}^{N_{\text{mel}}}$ is given by

$$\mathbf{z}_{\text{ref}} = 10\log_{10}(\mathbf{M}\mathbf{X}^2_{\text{ref}}), \tag{7}$$

where $\mathbf{M} \in \mathbb{R}^{N_{\text{mel}} \times (N_{\text{FFT}}/2+1)}$ represents the mel-scale filterbank, which compresses the non-redundant frequency bins into fewer mel-scaled bins N_{mel}. The

DNN is trained to predict the features $\mathbf{z}_{\text{ref}}$, such that $E_\theta(\mathbf{y}) = \hat{\mathbf{z}} \approx \mathbf{z}_{\text{ref}}$.

Once we have obtained $\hat{\mathbf{z}}$, the estimated time-averaged PSD in the original frequency domain can be computed by reversing the mel transformation:

$$\hat{\mathbf{X}}^2_{\text{ref}} = \mathbf{M}^T 10^{\hat{\mathbf{z}}/10}. \tag{8}$$

Here, $\mathbf{M}^T$ is the transpose of the mel filterbank, which interpolates the full FFT frequency resolution from the lower-dimensional mel representation.

3.2 Blind Parameter Estimation

The DNN model E_θ is designed to estimate the log-mel spectral features $\mathbf{z}(\mathbf{x})$ of an original recording $\mathbf{x}$, given a processed observation $\mathbf{y}$. The target parameters $\mathbf{z}(\mathbf{x})$ are extracted from the clean signal $\mathbf{x}$ using a procedure described in Sec. 3.1. During training, the observation $\mathbf{y}$ is generated by applying a forward transformation $f(\mathbf{x})$ to the same original recording. This transformation introduces a range of perturbations, including randomized spectral coloration, variations in loudness, room reverberation, and additive noise. This makes the spectral parameter estimation non-trivial.

We hypothesize that $\mathbf{z}(\mathbf{x})$, being both time-averaged and compressed through a mel filterbank, is correlated with the type of content in the input signal. For example, orchestral music tends to have a distinct spectral envelope, while speech has a different one, and characteristics like gender or age may further affect the spectral shape of speech signals. Based on this, the model $E_\theta(\mathbf{y})$ is expected to extract semantic information from the processed signal $\mathbf{y}$ and map it to the corresponding log-mel spectral features $\hat{\mathbf{z}}$.

To achieve this, we propose factorizing the DNN model E_θ into two components, such that

$$E_\theta(\mathbf{y}) = (\text{MLP}_{\theta_{\text{MLP}}} \circ \text{CLAP}_{\theta_{\text{CLAP}}})(\mathbf{y}), \tag{9}$$

where $\text{CLAP}_{\theta_{\text{CLAP}}}$ is a pretrained audio encoder [14], and $\text{MLP}_{\theta_{\text{MLP}}}$ is a small multi-layer perceptron (MLP) that learns the mapping from Contrastive Audio Language Pretraining (CLAP) embeddings to log-mel spectral features. CLAP is a joint audio-text encoder trained via contrastive learning, mapping audio and text pairs into a shared embedding space. We use the audio encoder of CLAP, with its parameters frozen during training, to extract semantic features from the signal $\mathbf{y}$.

Previous work has shown that embeddings from CLAP and similar models often fail to capture the details of audio effects and transformations, such as those introduced by spectral processing, limiting their sensitivity to these changes [15, 16]. This is beneficial in our case, as it suggests that the forward operator $f(\cdot)$, which introduces spectral coloration, may not significantly affect CLAP-derived embeddings. Thus, $\text{MLP}_{\theta_{\text{MLP}}}$ can focus on learning the relationship between the extracted embeddings and the log-mel spectral features.

The parameters of $\mathbf{z}(\mathbf{x})$ typically exhibit varying magnitudes, which generally decrease logarithmically as a function of frequency. Normalizing both the input and target data is beneficial during neural network training, as it ensures consistent scaling and improves model convergence. To achieve uniform error impact across all frequency bands, we first compute the average $\mathbf{z}_{\text{avg}}$ coefficients from a representative subset of the training data. The model is then trained to estimate the deviation of each instance from this average, ensuring balanced attention across the different frequency components.

The parameter estimation model E_θ is optimized using the following loss function:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}\sim p_x, f(\cdot)\sim p_f}\left[\|E_\theta(f(\mathbf{x})) - (\mathbf{z}(\mathbf{x}) - \mathbf{z}_{\text{avg}})\|_1\right], \tag{10}$$

where $\mathbf{x}$ represents clean audio data sampled from a distribution p_x (e.g., a dataset of recordings), and the forward operator $f(\cdot)$ is sampled from a distribution p_f, representing different types of data augmentations. As illustrated in Fig. 3, these degradations include room reverberation, randomized spectral coloration, additive noise, and gain randomization. Importantly, since the CLAP encoder is frozen during training, only the parameters of the MLP $\theta_{\text{MLP}} \subset \theta$ are optimized based on the loss objective. It is important to note that, due to the ℓ_1-norm minimization objective, at convergence the network E_θ approximates the conditional median of $\mathbf{z}(\mathbf{x})$ given the transformed input $f(\mathbf{x})$ under the training data distribution. Thus, the estimated power spectrum $\hat{\mathbf{z}} = E_\theta(f(\mathbf{x}))$ can be interpreted as

$$\hat{\mathbf{z}} \approx \text{median}_{\mathbf{z}\sim p(\mathbf{z}|f(\mathbf{x}))}[\mathbf{z} - \mathbf{z}_{\text{avg}}] + \mathbf{z}_{\text{avg}}, \tag{11}$$

where the median is computed implicitly over the target distribution defined by the training dataset.

The training process is illustrated in Fig. 1, and the signal processing conducted at inference time is summarized in Fig. 2. For simplicity, the detail of $\mathbf{z}_{\text{avg}}$ is omitted in both Fig. 1 and 2.

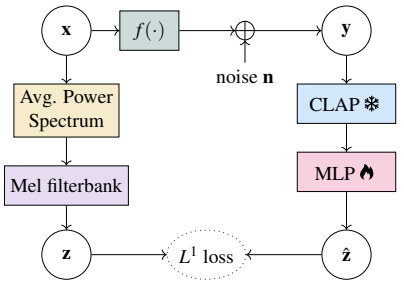


Fig. 1: Training diagram.

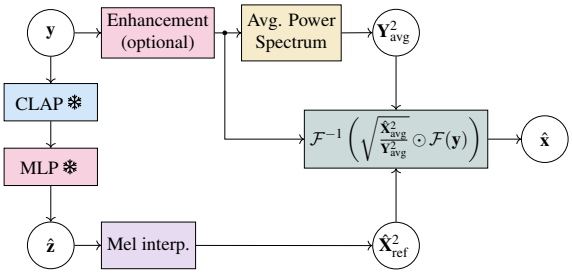


Fig. 2: Inference diagram.

4 Experiments

4.1 Training Data Pipeline

We develop a single model capable of processing both speech and music. The model is designed to exhibit robustness to background noise, gain variations, and reverberation, necessitating the inclusion of these disturbances within the training data pipeline. We utilize the speech datasets VCTK [17] and EARS [18], both of which consist of high-quality studio speech recordings sampled at 48 kHz, and which overall contain around 200 different speakers.

We choose MedleyDB [19] and DSD100 [20] as the music datasets, both offering studio-quality recordings across various genres such as rock, pop, classical, and jazz. These datasets are mixed by recording engineers, allowing us to consider the spectral balance as an appropriate target. Although the datasets provide isolated stems for every source in the microphone, we limit our use to the mixed tracks. Every audio segment undergoes loudness normalization to achieve a standardized volume level of -18 LUFS. This procedure guarantees a consistent loudness target, ensuring uniformity across different audio files.

The degradation process is outlined in Fig. 3, illustrating the practical implementation of $f(\cdot)$ as shown in Fig. 1. Initially, an audio segment with approximately 7 s in duration, containing either speech or music, is convolved with a room impulse response (RIR). We specifically use RIRs from the Arni dataset [21]. Subsequently, the audio is subjected to an equalization filter featuring random gains across various frequency bands. This filter is constructed in the frequency domain as a zero-phase FIR filter. The gain for a set of 26 bark-scaled frequency bands is stochastically sampled on a log-uniform scale ranging from -20 to 20 dB. In the next phase, noise is introduced. For noise sources, the TAU Urban Acoustic Scenes dataset [22] is employed during training. This dataset is integrated with speech and music by mixing with noise from this dataset, achieving an SNR that is log-uniformly sampled between 5 and 50 dB.

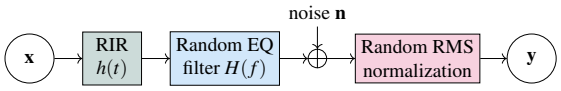


Fig. 3: Pipeline illustrating the applied degradations.

Finally, the gain of the mixture is adjusted through a random root-mean-square (RMS) normalization. The RMS normalization is implemented as $y = (\alpha\sqrt{L}/\|\mathbf{x}\|_2)\mathbf{x}$, with the parameter α being uniformly sampled within -5 dB and 30 dB.

4.2 Training Details

The training of the model consisted of conducting 100,000 iterations, utilizing the Adam optimization algorithm. The learning rate employed during this process was set to 1×10^{-4}, and the values of the beta momentum parameters were configured to be (0.9, 0.999). For the training data, audio segments of 7 s sampled at 48 kHz were used. The training was performed with a batch size of 16.

We chose a number of mel-frequency coefficients $N_{\text{mel}} = 32$, extracted using the STFT with FFT size 2048, a Hann window with size 1024, and a hop length of 256 samples. The employed MLP contains 0.34 M parameters and is based on three linear layers with a hidden size of 256 and Leaky-ReLU nonlinearities.

4.3 Evaluation Data

The model is evaluated on speech and music, and in different scenarios, according to the applied degradations.

We distinguish the following scenarios: 1. Speech equalization, 2. Music equalization, 3. Noisy speech equalization, and 4. Reverberant speech equalization.

For scenarios 1, 3, and 4, we evaluate on VCTK speakers "p351", "p360"–"p364", "p374", "p376" and EARS speakers "p100"–"p107". In scenario 2, we extract 8-s segments from separated test splits of MedleyDB and DSD100. For each of these scenarios, a test set comprising 1,000 samples is created. To each example, a random filter is applied following the same procedure used during training, and the gain is randomized with an RMS in the interval [0.005, 0.30], extending the range seen at training time.

In scenario 3, we introduce noise from the DEMAND dataset [23] and apply SNRs ranging between -5 dB and 30 dB, which also extends beyond the training range. Finally, in scenario 4, the signals are convolved with RIRs, utilizing simulated RIRs from pyroomacoustics [24], with a reverberation time T_{60} spanning [0.1, 1.0]. In both scenarios, we apply DeepFilterNet2 [25] to pre-process audio signals that are either noisy or reverberant. DeepFilterNet2 was selected due to its open-source availability, lightweight architecture, high processing speed, and acceptable performance. It is important to note that the original signal remains accessible to the model, while the pre-processing is integrated into the equalization pipeline, as shown in Fig. 2.

4.4 Metrics

To evaluate the performance of the model, we consider several metrics that provide a comprehensive analysis of its effectiveness in estimating the power spectrum and enhancing audio signals. These metrics capture both spectral accuracy and audio-domain fidelity.

We compute the L1 error between the target average power spectrum extracted from the reference high-quality recording and the estimated power spectrum. This metric corresponds directly to the loss function used during training, as defined in Eq. (10). It provides a measure of how closely the model's predictions align with the target power spectrum, reflecting its ability to learn accurate spectral representations.

To evaluate the fidelity of the equalized audio signal, we employ the Log-Spectral Distortion (LSD), which measures differences in spectral features between the reference and predicted audio signals. The LSD is computed as

$$\mathrm{LSD} = \frac{1}{MK} \sum_{m,k} \left| \log(|\mathcal{F}(\mathbf{x})| + \varepsilon) - \log(|\mathcal{F}(\hat{\mathbf{x}})| + \varepsilon) \right|, \tag{12}$$

where $\mathcal{F}$ represents the Short-Time Fourier Transform (STFT) operator, M is the number of time frames, K is the number of frequency bins, and ε is a small constant (e.g., 10^{-6}) to avoid logarithmic instabilities.

While these objective metrics are useful for evaluating model performance, they are not always directly indicative of perceptual quality. For example, the equalization applied to music examples may deviate from the median power spectrum the model predicts. This is because the model's estimation inherently reflects the statistical properties of the training data, whereas real-world audio signals often exhibit unique spectral characteristics. Thus, while the metrics provide quantitative insights, qualitative evaluation through listening tests is essential to fully assess the method's impact on perceptual audio quality.

4.5 Baselines

To contextualize the performance of our proposed method, we designed and evaluate the proposed method against several baselines. The first baseline, which we call *Average*, computes the average power spectrum of the training data, and uses that to compute the inverse filter in Eq. (5). We compute the average of the speech data and music data separately. By using these fixed average spectra as predictions, this baseline is meant to highlight the effectiveness of the proposed method at leveraging data-specific patterns compared to a naive global estimate.

The second baseline, which we call *End-to-end*, replaces the frozen CLAP encoder with a fully trainable encoder, allowing the model to jointly optimize feature extraction and power spectrum estimation. We used the same neural network architecture as in the setup from Steinmetz et al. [9]. This baseline aims to demonstrate the advantages of using a pre-trained representation over task-specific feature learning.

Finally, the *Oracle* baseline uses parameters directly estimated from the reference high-quality audio signal, bypassing the need for a model prediction. This serves as an upper bound on performance within our framework, illustrating the best possible outcome achievable when the target parameters are perfectly known.

5 Objective Evaluation

5.1 Speech Equalization and Gain Adjustment

In this scenario, speech signals undergo arbitrary EQ and random RMS processing. The outcomes are depicted in Fig. 4. Each plot displays the median curves represented by colored lines, with the InterQuartile Range (25%-75%) illustrated as shaded regions. Figure 4a illustrates L^1 errors on parameter estimation in relation to frequency, highlighting increased errors in higher frequency ranges. Both the Proposed and End-to-end models display comparable performance, surpassing the Average model. Figure 4b presents L^1 errors concerning the RMS of the altered signal, showing consistent performance across all RMS levels, despite training only within the [0.02, 0.18] interval. Performance seems similar even beyond this range. Figure 4c depicts LSD relative to frequency, indicating that Proposed, Average, and End-to-end perform similarly, though less effectively than the Oracle condition. In Figure 4d, LSD errors in relation to RMS reveal higher errors at lower RMS values, even in the Oracle scenario, underscoring a fundamental limitation of the employed inverse filter approach (see Eq. (5)).

5.2 Music Equalization and Gain Adjustment

In Scenario 2, musical signals were processed utilizing randomized equalization filters and gain levels. The outcomes, illustrated in Fig. 5 and mirroring the data from Fig. 4, exhibit patterns akin to those identified in Scenario 1. Nevertheless, here, both the Average and End-to-end baselines demonstrate significant performance reduction across all metrics. This deterioration in the Average baseline's effectiveness is probably attributable to the wider variability found in music as opposed to speech. The Proposed model sustains superior overall performance compared to the End-to-end baseline, suggesting that the use of pretrained CLAP embeddings facilitates robust feature extraction, even for the more complex music signals. This scenario is also evaluated through subjective listening in Sec. 6.

5.3 Noisy Speech Equalization

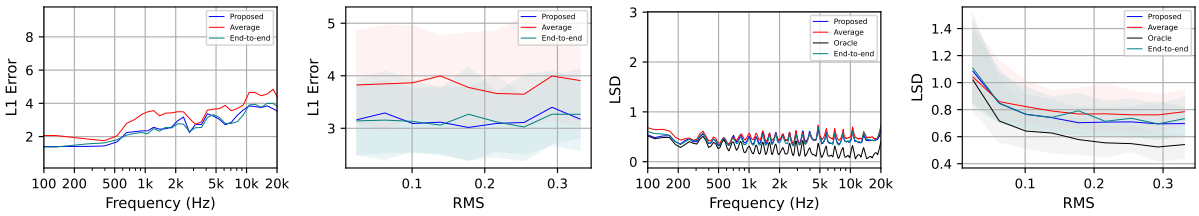
This scenario investigates the impact of background noise on inverse filter estimation by analyzing the relationship between the objective metrics and the input SNR. Figure 6a presents the L^1 error of the estimated parameters as a function of SNR. Both the Proposed and End-to-end methods exhibit similar performance, with a noticeable but relatively minor increase in error at lower SNRs, which aligns with expectations given the added noise.

The LSD metric, defined between audio waveforms, is affected by the speech enhancement model used for preprocessing. In Fig. 6b, we evaluate performance using DeepFilterNet2 [25] for speech enhancement prior to equalization. Here, all methods exhibit similar trends, heavily influenced by the SNR. To explore an idealized scenario, we apply the estimated equalization (derived from noisy measurements) to a noiseless version of the degraded signal, simulating perfect denoising while retaining spectral coloration. As shown in Fig. 6c, this setup significantly improves LSD performance, achieving near-uniform results across the SNR range. In this case, the *Proposed* and *Average* conditions perform comparably, while the *End-to-end* baseline shows a noticeable decline in performance.

5.4 Reverberant Speech Equalization

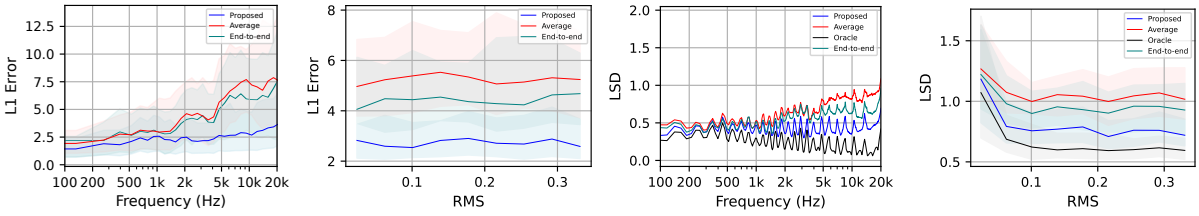
This scenario builds upon the analysis in Sec. 5.3, shifting the focus to robustness against reverberation. We examine how different metrics vary with the reverberation time applied to the measurements. Figure 7a shows the L^1 parameter estimation error as a function of the reverberation time, T_{60}. The Proposed and End-to-end methods perform similarly, with a slight increase in error at longer T_{60} values.

Figure 7b depicts the audio-domain LSD metric with respect to T_{60}, where the equalization is applied to signals enhanced by DeepFilterNet2. As in Sec. 5.3, the metrics are heavily influenced by the performance of the speech enhancement model, overshadowing the underlying robustness of the equalization methods. To address this limitation, we repeat the experiment under an idealized scenario where perfect dereverberation is assumed. Specifically, the reference dry signal is equalized using the inverse filter estimated from the reverberant measurement. The result, presented in Fig. 7c, shows that the correlation between the metrics and T_{60} becomes negligible under these conditions. In this idealized setup, the Proposed method outperforms End-to-end and demonstrates a marginal but consistent advantage over the Average method in terms of LSD, further highlighting its robustness to reverberation.



(a) L_1 error vs frequency **(b)** L_1 error vs RMS **(c)** LSD error vs frequency **(d)** LSD error vs RMS

Fig. 4: Objective metrics from Scenario 1: Equalization and gain adjustment for speech.



(a) L_1 error vs frequency **(b)** L_1 error vs RMS **(c)** LSD vs frequency **(d)** LSD vs RMS

Fig. 5: Objective metrics from Scenario 2: Equalization and gain adjustment for music.

6 Subjective Evaluation

The goal of the listening test was to evaluate the performance of the proposed method for music equalization, focusing on subjective preference. As music equalization is inherently subjective, we aimed to determine which type of equalization participants preferred. The primary question for the test was simple: "Which one do you prefer?". Participants were encouraged to choose randomly when they were unsure.

The test followed a pairwise comparison preference method, where participants were presented with two audio samples and asked to choose the one they preferred. This allowed for direct comparisons between different equalization conditions. We selected four music excerpts, each approximately 7 s long, from four different genres: *Jazz*, *Opera*, *Pop*, and *Rock*. These samples are available in the attached material.

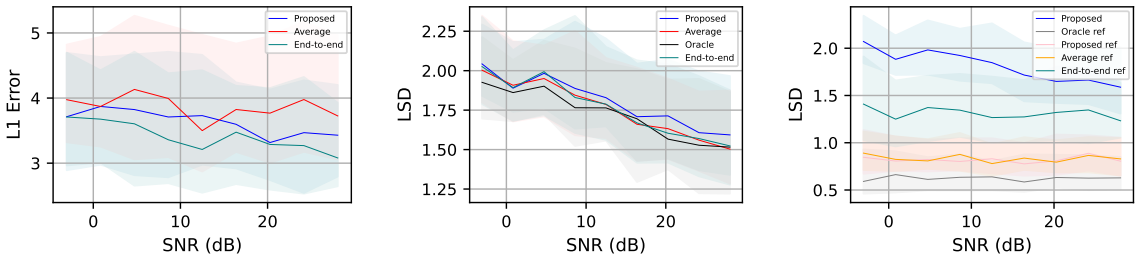
Participants assessed five distinct scenarios: the *Reference* condition, featuring the untouched audio; the *Distorted* condition, where the original audio sample was pre-processed with a random equalization filter before serving as input for subsequent methods; the *Proposed* condition, which was treated using the technique suggested in this study; the *Oracle* condition, which utilized the same equalization approach as the *Proposed* but with access to the reference for calculating the average power spectra; and the *Average* condition, which employed an average equalization approach based on the spectral average of the music training dataset. All samples were loudness-normalized preceding the tests to ensure that evaluations focused on spectral coloration rather than differences in loudness. Every pairwise comparison was conducted twice to eliminate potential biases that might occur if users tend to favor the letters A or B when uncertain. Additionally, all pages were displayed in a random sequence.

We implemented the test using the WebMUSHRA tool [26], with an unofficial version adapted specifically for preference tests [1]. The test involved 8 participants, who all completed it in less than 15 min. The test was intentionally kept brief to prevent participant fatigue. All participants had prior experience with listening tests. The average age was 28, with an equal gender distribution. No participants were excluded.

6.1 Listening Test Results

The counts are aggregated across all participants and trials. The combined results for all genres are provided in Table 1. Figure 8(e) also summarizes the aggregated results of the pairwise comparison of the proposed method with the other conditions. The tables show, for each possible pair, the number of wins, i.e., the times a participant chose one over the other. These tables illustrate, for each possible pair of conditions, the number of times participants preferred one option

[1] https://github.com/Simon-Stone/webMUSHRA/tree/preference_test

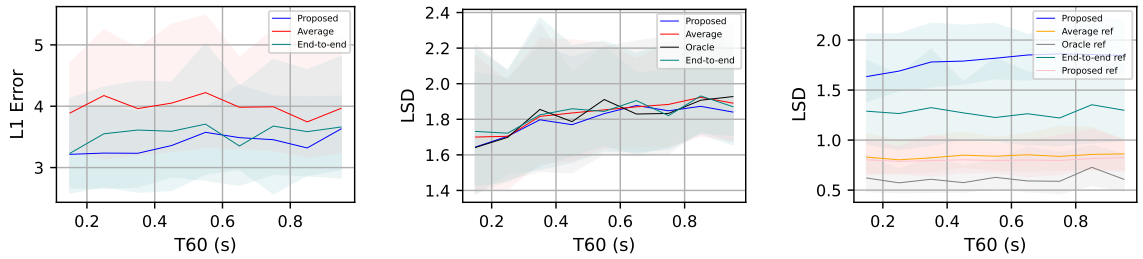


(a) L_1 error vs SNR

(b) LSD error vs SNR. Using speech enhancement.

(c) LSD error vs SNR. Using denoised reference.

Fig. 6: Objective metrics from Scenario 3: Equalization and gain adjustment for noisy speech



(a) L_1 error vs reverberation time T_{60}

(b) LSD vs reverberation time T_{60}. Using speech enhancement.

(c) LSD vs reverberation time T_{60}. Using dereverberated reference.

Fig. 7: Objective metrics from Scenario 4: Equalization and gain adjustment for reverberant speech

over the other. The results from the *Jazz*, *Opera*, *Pop* and *Rock* examples are represented in Table 2. Figures 8(a–d) also show the results of the comparison of the proposed method with the other four conditions.

For each pair, a statistical analysis was performed to determine whether the observed preference was significant. Specifically, a Z-test for proportions was conducted, comparing the proportion of times each condition was selected. The Z-statistic quantifies the difference between these proportions, taking into account the sample size and variability. A positive Z-statistic indicates that the first condition (e.g., “Option A") was preferred more often than the second (e.g., “Option B"), while a negative Z-statistic suggests the opposite. The p-value associated with the Z-statistic represents the probability that the observed difference in preferences could occur due to random chance. A p-value below 0.01 is typically considered statistically significant, suggesting that participants had a genuine preference rather than random variability. The tables use color coding to highlight statistically significant results. The condition preferred more frequently in significant comparisons is highlighted in green as the winner, while the less-preferred condition is highlighted in red.

Table 1: Aggregated preference listening test results across all genres. In the statistically significant cases ($P < 0.01$), the winning and losing methods are indicated with green and red font, respectively.

Option A	Option B	Wins A	Wins B	Z-Statistic	P-Value
proposed	average	51	13	6.72	1.85e-11
proposed	oracle	31	33	-0.35	0.72
proposed	distorted	60	4	9.90	4.18e-23
proposed	reference	15	49	-6.01	1.85e-09
oracle	average	40	24	2.83	0.0047
oracle	distorted	58	6	9.19	3.84e-20
oracle	reference	15	49	-6.01	1.85e-09
average	distorted	55	9	8.13	4.23e-16
average	reference	8	56	-8.49	2.15e-17
distorted	reference	6	58	-9.19	3.84e-20

6.2 Discussion

The results provide meaningful insights into the performance of the tested methods across all genres. As expected, Table 1 shows that the distorted condition was always rated the lowest, underscoring the detrimental impact of the applied random EQ filter on perceived audio quality. The proposed method outperformed or performed comparably to the average condition, with

Table 2: Results of the preference test across various genres. For brevity, paired comparisons between conditions other than the proposed method are omitted.

Genre	Option A	Option B	Wins A	Wins B	Z-Statistic	P-Value
Jazz	proposed	average	8	8	0.00	1.00
	proposed	oracle	8	8	0.00	1.00
	proposed	distorted	16	0	5.66	1.54e-08
	proposed	reference	5	11	-2.12	0.03
Opera	proposed	average	13	3	3.54	0.0004
	proposed	oracle	12	4	2.83	0.0047
	proposed	distorted	16	0	5.66	1.54e-08
	proposed	reference	7	9	-0.71	0.48
Pop	proposed	average	14	2	4.24	2.21e-05
	proposed	oracle	7	9	-0.71	0.48
	proposed	distorted	15	1	4.95	7.43e-07
	proposed	reference	2	14	-4.24	2.21e-05
Rock	proposed	average	16	0	5.66	1.54e-08
	proposed	oracle	4	12	-2.83	0.0047
	proposed	distorted	13	3	3.54	0.0004
	proposed	reference	1	15	-4.95	7.43e-07

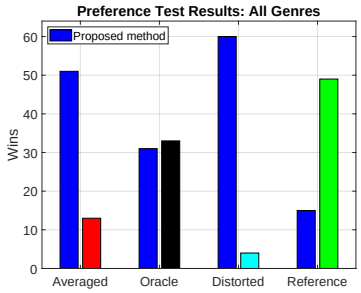


Fig. 8: Listening test results from pairwise comparison of the proposed method. The data correspond to the Wins A and B in the four first lines of Table 1.

Jazz being an exception where the proposed method and the average condition yielded similar results, as can be seen in Table 2. This highlights the capability of the proposed method to retrieve data patterns.

When compared to the *Oracle* condition, the proposed method demonstrated comparable results in the aggregated analysis of Table 1, with no significant differences observed. This is particularly noteworthy because it suggests that the blind power spectrum estimator employed in the proposed method nearly reaches the performance bound under the given conditions. Interestingly, in the *Opera* genre, the proposed method outperformed the Oracle condition, see Table 2. This outcome may stem from the fact that the median estimate of the average power spectrum (see Eq. (11)), used in the proposed method, is subjectively more pleasing than the one derived directly from the reference.

However, both the proposed method and the Oracle condition were consistently rated lower than the Reference condition, as seen in Table 1. This points to a limitation of the inverse filter approximation. The *Reference* condition serves as an idealized target, and the observed quality gap between the Oracle method and the Reference condition suggests that further refinement is needed. One promising avenue for improvement could involve increasing the resolution of the estimated coefficients by expanding the number of frequency bands in the set of parameters that the model estimates.

7 Conclusions

This paper introduces a method for automatic audio equalization using blind power spectrum estimation combined with inverse filtering, showcasing reliable performance across various challenging conditions. The approach has been designed to work with both speech and music and to be robust to noise and reverberation. The proposed method shows a distinct advantage when processing audio with varied semantic content, such as music, while its benefit over baseline methods is less pronounced for speech. Our listening test results indicate a marked preference for this method in different music genres, achieving results close to Oracle level without the need for reference data. However, the gap between the proposed method and the Reference baseline highlights opportunities for further improvement. Current limitations are primarily linked to the inverse filtering stage, particularly the use of a subsampled spectrum via a mel-scale filterbank.

8 Acknowledgment

This work has been financed in part by Nokia Technologies through the DeepMix project (Aalto University project no. 411116).

References

[1] V. Välimäki and J. D. Reiss. All about audio equalization: Solutions and frontiers. *Applied Sciences*, 6(5):129, 2016.

[2] B. De Man, J. Reiss, and R. Stables. Ten years of automatic mixing. In *Proc. 3rd Workshop on Intelligent Music Production*, Salford, UK, 2017.

[3] M. A. Martínez-Ramírez et al. Automatic music mixing with deep learning and out-of-domain data. In *Proc. ISMIR*, 2022.

[4] T. Sankowsky-Rothe, M. Blau, S. Köhler, and A. Stirnemann. Individual equalization of hearing aids with integrated ear canal microphones. *Acta Acustica united with Acustica*, 101(3):552–566, 2015.

[5] X. Liu, J. D. Reiss, and A. Mourgela. An automatic mixing system for teleconferencing. In *Proc. Audio Engineering Society Conv. 153*, 2022.

[6] T.G. Stockham, T.M. Cannon, and R.B. Ingebretsen. Blind deconvolution through digital signal processing. *Proc. IEEE*, 63(4):678–692, 1975.

[7] Z. Ma, J. D. Reiss, and D. Black. Implementation of an intelligent equalization tool using Yule-Walker for music mixing and mastering. In *Proc. Audio Engineering Society Conv. 134*, page 5, 2013.

[8] M. A. Martínez-Ramírez and J. D. Reiss. End-to-end equalization with convolutional neural networks. In *Proc. International Conference on Digital Audio Effects (DAFx)*, 2018.

[9] C. J. Steinmetz, N. J. Bryan, and J. D. Reiss. Style transfer of audio effects with differentiable signal processing. *Journal of the Audio Engineering Society*, 70(9):708–721, 2022.

[10] C. Peladeau and G. Peeters. Blind estimation of audio effects using an auto-encoder approach and differentiable digital signal processing. In *Proc. IEEE ICASSP*, pages 856–860, 2024.

[11] C. J. Steinmetz et al. ST-ITO: Controlling audio effects for style transfer with inference-time optimization. In *Proc. ISMIR*, 2024.

[12] F. Mockenhaupt, J. S. Rieber, and S. Nercessian. Automatic equalization for individual instrument tracks using convolutional neural networks. In *Proc. International Conference on Digital Audio Effects (DAFx)*, 2024.

[13] E. Moliner et al. A diffusion-based generative equalizer for music restoration. In *Proc. International Conference on Digital Audio Effects (DAFx)*, 2024.

[14] B. Elizalde, S. Deshmukh, and H. Wang. Natural language supervision for general-purpose audio representations. In *Proc. IEEE ICASSP*, pages 336–340, 2024.

[15] A. Gui et al. Adapting Frechet audio distance for generative music evaluation. In *Proc. IEEE ICASSP*, pages 1331–1335, 2024.

[16] S. H. Hawley and C. J. Steinmetz. Leveraging neural representations for audio manipulation. In *Proc. Audio Engineering Society Conv.*, 2023.

[17] C. Valentini-Botinhao et al. Reverberant speech database for training speech dereverberation algorithms and TTS models. *Univ. Edinburgh*, 2016.

[18] J. Richter et al. EARS: An anechoic fullband speech dataset benchmarked for speech enhancement and dereverberation. In *Proc. INTERSPEECH*, 2024.

[19] R. M. Bittner et al. Medleydb 2.0: New data and a system for sustainable data collection. *ISMIR Late Breaking and Demo Papers*, 36, 2016.

[20] A. Liutkus et al. The 2016 signal separation evaluation campaign. In *Proc. Latent Variable Analysis and Signal Separation: 13th International Conference, LVA/ICA*, pages 323–332, 2017.

[21] K. Prawda, S. J. Schlecht, and V. Välimäki. Calibrating the Sabine and Eyring formulas. *J. Acoust. Soc. Amer.*, 152(2):1158–1169, 2022.

[22] T. Heittola, A. Mesaros, and T. Virtanen. TAU Urban Acoustic Scenes 2022 Mobile, Development dataset, March 2022.

[23] J. Thiemann, N. Ito, and E. Vincent. The diverse environments multi-channel acoustic noise database (demand): A database of multichannel environmental noise recordings. In *Proc. Meetings on Acoustics*, volume 19, 2013.

[24] R. Scheibler, E. Bezzam, and I. Dokmanić. Pyroomacoustics: A Python package for audio room simulation and array processing algorithms. In *Proc. IEEE ICASSP*, pages 351–355, 2018.

[25] H. Schröter et al. DeepFilterNet2: Towards real-time speech enhancement on embedded devices for full-band audio. In *Proc. International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 1–5, 2022.

[26] M. Schoeffler et al. WebMUSHRA—A comprehensive framework for web-based listening tests. *Journal of Open Research Software*, 6(1), 2018.